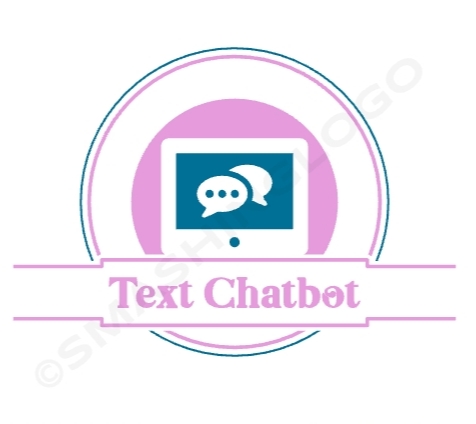
**CLOUD APPLICATION DEVELOPMENT:**

**Phase 5:**

**Project Title: Chatbot Deployment with IBM Cloud Watson Assistant.**

In our chatbot, we used html, css and javascript for developing our chatbot.

The logo of our text chatbot be,



The implementation of our text chatbot be,

**CODE:**

<DOCTYPE html>

<html lang="en">

<head>

<title>Create Your Own Chatbot</title>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width" />

<link rel="stylesheet" href="styles.css" />

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:400,700,300">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.1.2/css/material-design-iconic-font.min.css">

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Epilogue:wght@400&family=Finger+Paint&display=swap">

</head>

<style>

\* {

font-size: 1.3vw;

font-family: "Epilogue", italic;

}

html {

--scrollbarBG: #fff;

--thumbBG: #90a4ae;

}

body {

background: cyan;

}

body .card {

height: 45vw;

width: 35vw;

background-color: purple;

margin-left: 30vw;

margin-top: 5vw;

box-shadow: 2vw 2vw 12vw 3vw #ccc;

}

body .card #header {

height: 5vw;

background: #000;

padding: 0vw;

}

body .card #header h1 {

color: #fff;

font-size: 2vw;

font-family: "Finger Paint", cursive;

padding: 1vw;

}

body .card #message-section::-webkit-scrollbar {

width: 10px;

}

body .card #message-section {

height: 32vw;

padding: 0 2.5vw;

overflow-y: auto;

scrollbar-width: thin;

scrollbar-color: var(--thumbBG) var(--scrollbarBG);

}

body .card #message-section::-webkit-scrollbar-track {

background: var(--scrollbarBG);

}

body .card #message-section::-webkit-scrollbar-thumb {

background-color: var(--thumbBG);

border-radius: 6px;

border: 3px solid var(--scrollbarBG);

}

body .card #message-section #bot,

body .card #message-section #user {

position: relative;

bottom: 0;

min-height: 1.5vw;

border: 0.15vw solid #777;

background-color: #fff;

border-radius: 0px 1.5vw 1.5vw 1.8vw;

padding: 1vw;

margin: 1.5vw 0;

}

body .card #message-section #user {

border: 1.5px solid #000;

border-radius: 1.5vw 0vw 1.5vw 1.8vw;

background-color: #000;

float: right;

}

body .card #message-section #user #user-response {

color: #fff;

}

body .card #message-section .message {

color: #000;

clear: both;

line-height: 1.2vw;

font-size: 1.2vw;

padding: 8px;

position: relative;

margin: 8px 0;

max-width: 85%;

word-wrap: break-word;

z-index: 2;

}

body .card #input-section {

z-index: 1;

padding: 0 2.5vw;

display: flex;

flex-direction: row;

align-items: flex-end;

overflow: hidden;

height: 6vw;

width: 100%;

}

body .card #input-section input {

color: #000;

min-width: 0.5vw;

outline: none;

height: 5vw;

width: 26vw;

border-top: none;

border-left: none;

border-right: none;

border-bottom: solid #000 0.1vw;

}

body .card .send {

background: transparent;

border: 0;

cursor: pointer;

flex: 0 0 auto;

margin-left: 1.4vw;

margin-right: 0vw;

padding: 0;

position: relative;

outline: none;

}

body .card .send .circle {

position: relative;

width: 4.8vw;

height: 4.8vw;

display: flex;

align-items: center;

justify-content: center;

}

body .card .send .circle i {

font-size: 3vw;

margin-left: -1vw;

margin-top: 1vw;

}

</style>

<body>

<body>

<div class="card">

<div id="header">

<h1>Chatter box!</h1>

</div>

<div id="message-section">

<div class="message" id="bot"><span id="bot-response">Hello.. I'm listening! Go on..</span></div>

</div>

<div id="input-section">

<input id="input" type="text" placeholder="Type a message" autocomplete="off" autofocus="autofocus"/>

<button class="send" onclick="sendMessage()">

<div class="circle"><i class="zmdi zmdi-mail-send"></i></div>

</button>

</div>

</div>

<script>

const userMessage = [

["hi", "hey", "hello"],

["sure", "yes", "no"],

["are you genious", "are you nerd", "are you intelligent"],

["i hate you", "i dont like you"],

["how are you", "how is life", "how are things", "how are you doing"],

["how is corona", "how is covid 19", "how is covid19 situation"],

["what are you doing", "what is going on", "what is up"],

["how old are you"],

["who are you", "are you human", "are you bot", "are you human or bot"],

["who created you", "who made you", "who is your creator"],

[

"your name please",

"your name",

"may i know your name",

"what is your name",

"what call yourself"

],

["i love you"],

["happy", "good", "fun", "wonderful", "fantastic", "cool", "very good"],

["bad", "bored", "tired"],

["help me", "tell me story", "tell me joke"],

["ah", "ok", "okay", "nice", "welcome"],

["thanks", "thank you"],

["what should i eat today"],

["bro"],

["what", "why", "how", "where", "when"],

["corona", "covid19", "coronavirus"],

["you are funny"],

["i dont know"],

["boring"],

["im tired"]

];

const botReply = [

["Hello!", "Hi!", "Hey!", "Hi there!"],

["Okay"],

["Yes I am! "],

["I'm sorry about that. But I like you dude."],

[

"Fine... how are you?",

"Pretty well, how are you?",

"Fantastic, how are you?"

],

["Getting better. There?", "Somewhat okay!", "Yeah fine. Better stay home!"],

[

"Nothing much",

"About to go to sleep",

"Can you guess?",

"I don't know actually"

],

["I am always young."],

["I am just a bot", "I am a bot. What are you?"],

["Sabitha Kuppusamy"],

["I am nameless", "I don't have a name"],

["I love you too", "Me too"],

["Have you ever felt bad?", "Glad to hear it"],

["Why?", "Why? You shouldn't!", "Try watching TV", "Chat with me."],

["What about?", "Once upon a time..."],

["Tell me a story", "Tell me a joke", "Tell me about yourself"],

["You're welcome"],

["Briyani", "Burger", "Sushi", "Pizza"],

["Dude!"],

["Yes?"],

["Please stay home"],

["Glad to hear it"],

["Say something interesting"],

["Sorry for that. Let's chat!"],

["Take some rest, Dude!"]

];

const alternative = [

"Same here, dude.",

"That's cool! Go on...",

"Dude...",

"Ask something else...",

"Hey, I'm listening..."

];

const synth = window.speechSynthesis;

function voiceControl(string) {

let u = new SpeechSynthesisUtterance(string);

u.text = string;

u.lang = "en-aus";

u.volume = 1;

u.rate = 1;

u.pitch = 1;

synth.speak(u);

}

function sendMessage() {

const inputField = document.getElementById("input");

let input = inputField.value.trim();

input != "" && output(input);

inputField.value = "";

}

document.addEventListener("DOMContentLoaded", () => {

const inputField = document.getElementById("input");

inputField.addEventListener("keydown", function (e) {

if (e.code === "Enter") {

let input = inputField.value.trim();

input != "" && output(input);

inputField.value = "";

}

});

});

function output(input) {

let product;

let text = input.toLowerCase().replace(/[^\w\s\d]/gi, "");

text = text

.replace(/[\W\_]/g, " ")

.replace(/ a /g, " ")

.replace(/i feel /g, "")

.replace(/whats/g, "what is")

.replace(/please /g, "")

.replace(/ please/g, "")

.trim();

let comparedText = compare(userMessage, botReply, text);

product = comparedText

? comparedText

: alternative[Math.floor(Math.random() \* alternative.length)];

addChat(input, product);

}

function compare(triggerArray, replyArray, string) {

let item;

for (let x = 0; x < triggerArray.length; x++) {

for (let y = 0; y < replyArray.length; y++) {

if (triggerArray[x][y] == string) {

items = replyArray[x];

item = items[Math.floor(Math.random() \* items.length)];

}

}

}

//containMessageCheck(string);

if (item) return item;

else return containMessageCheck(string);

}

function containMessageCheck(string) {

let expectedReply = [

[

"Good Bye, dude",

"Bye, See you!",

"Dude, Bye. Take care of your health in this situation."

],

["Good Night, dude", "Have a sound sleep", "Sweet dreams"],

["Have a pleasant evening!", "Good evening too", "Evening!"],

["Good morning, Have a great day!", "Morning, dude!"],

["Good Afternoon", "Noon, dude!", "Afternoon, dude!"]

];

let expectedMessage = [

["bye", "tc", "take care"],

["night", "good night"],

["evening", "good evening"],

["morning", "good morning"],

["noon"]

];

let item;

for (let x = 0; x < expectedMessage.length; x++) {

if (expectedMessage[x].includes(string)) {

items = expectedReply[x];

item = items[Math.floor(Math.random() \* items.length)];

}

}

return item;

}

function addChat(input, product) {

const mainDiv = document.getElementById("message-section");

let userDiv = document.createElement("div");

userDiv.id = "user";

userDiv.classList.add("message");

userDiv.innerHTML = `<span id="user-response">${input}</span>`;

mainDiv.appendChild(userDiv);

let botDiv = document.createElement("div");

botDiv.id = "bot";

botDiv.classList.add("message");

botDiv.innerHTML = `<span id="bot-response">${product}</span>`;

mainDiv.appendChild(botDiv);

var scroll = document.getElementById("message-section");

scroll.scrollTop = scroll.scrollHeight;

voiceControl(product);

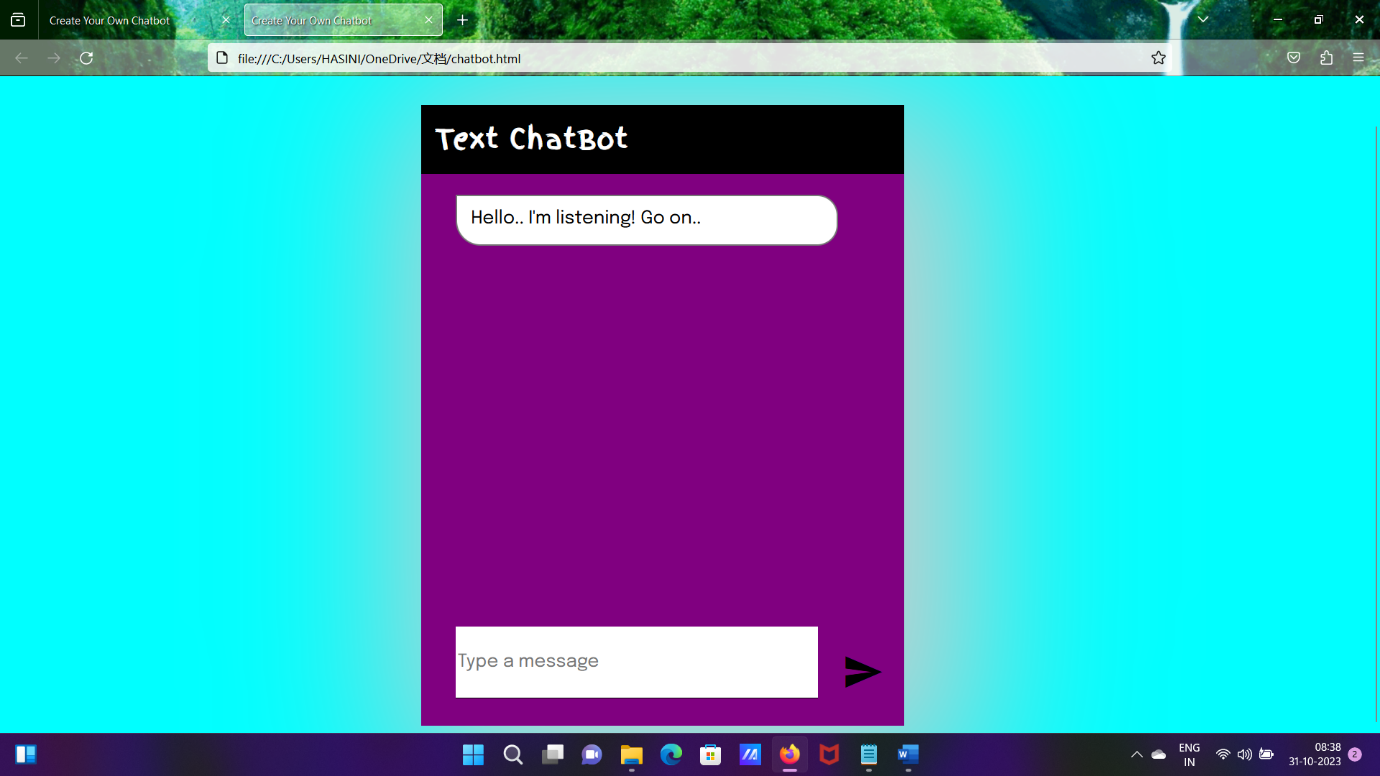
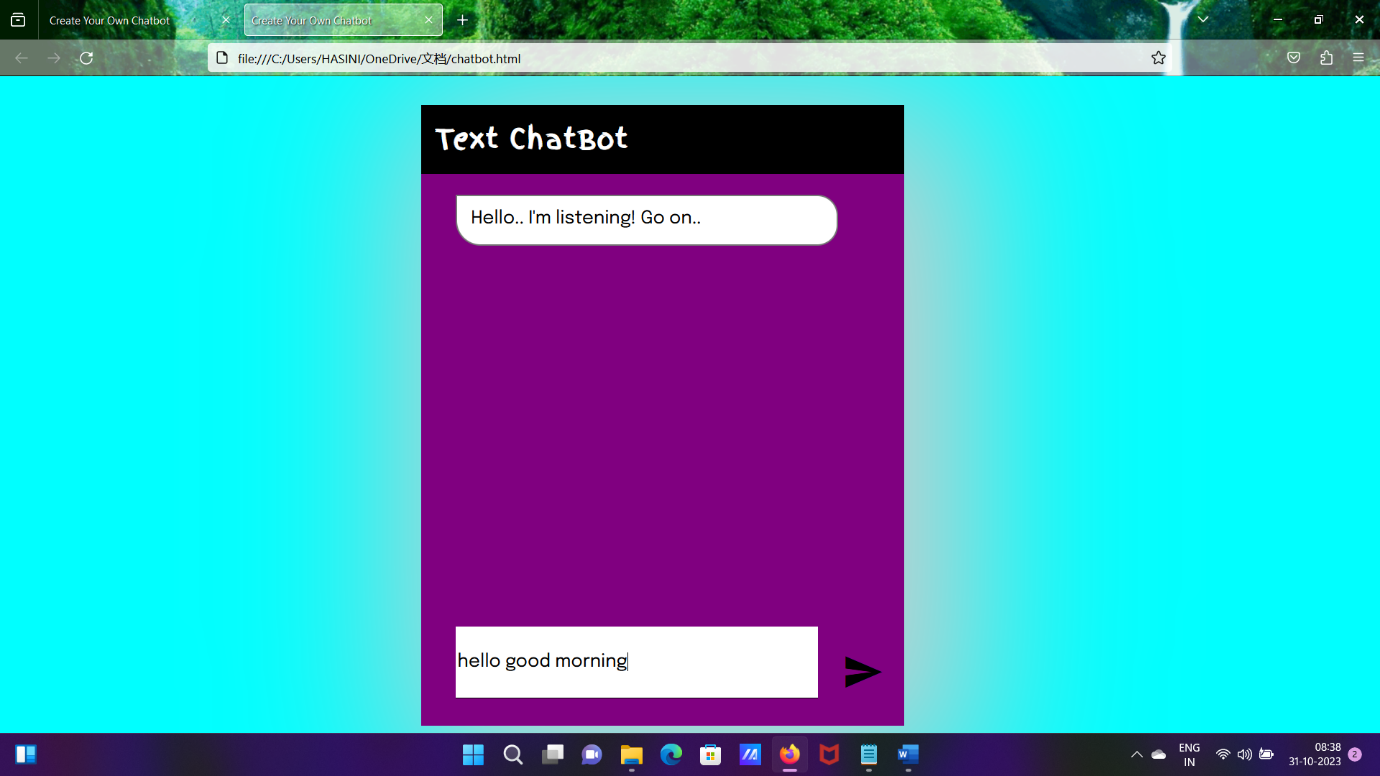
}

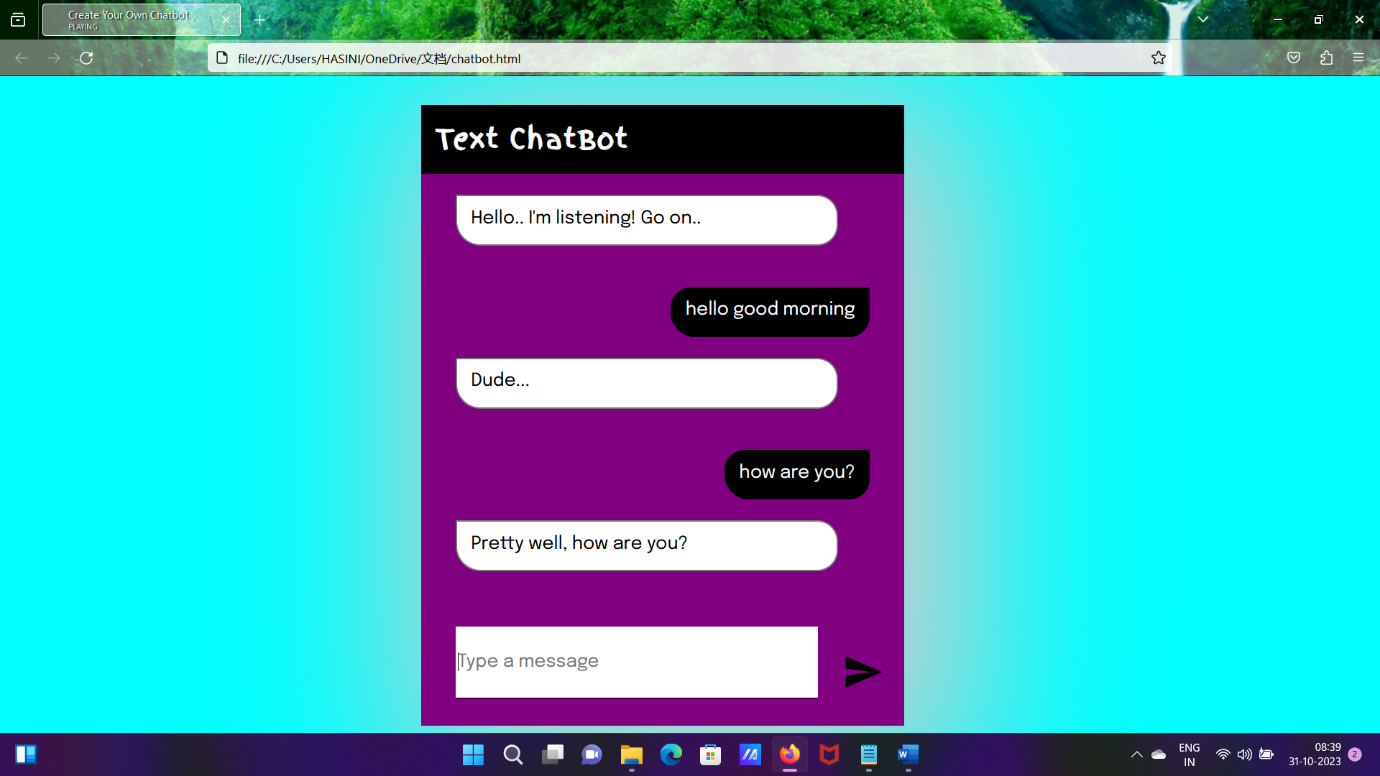
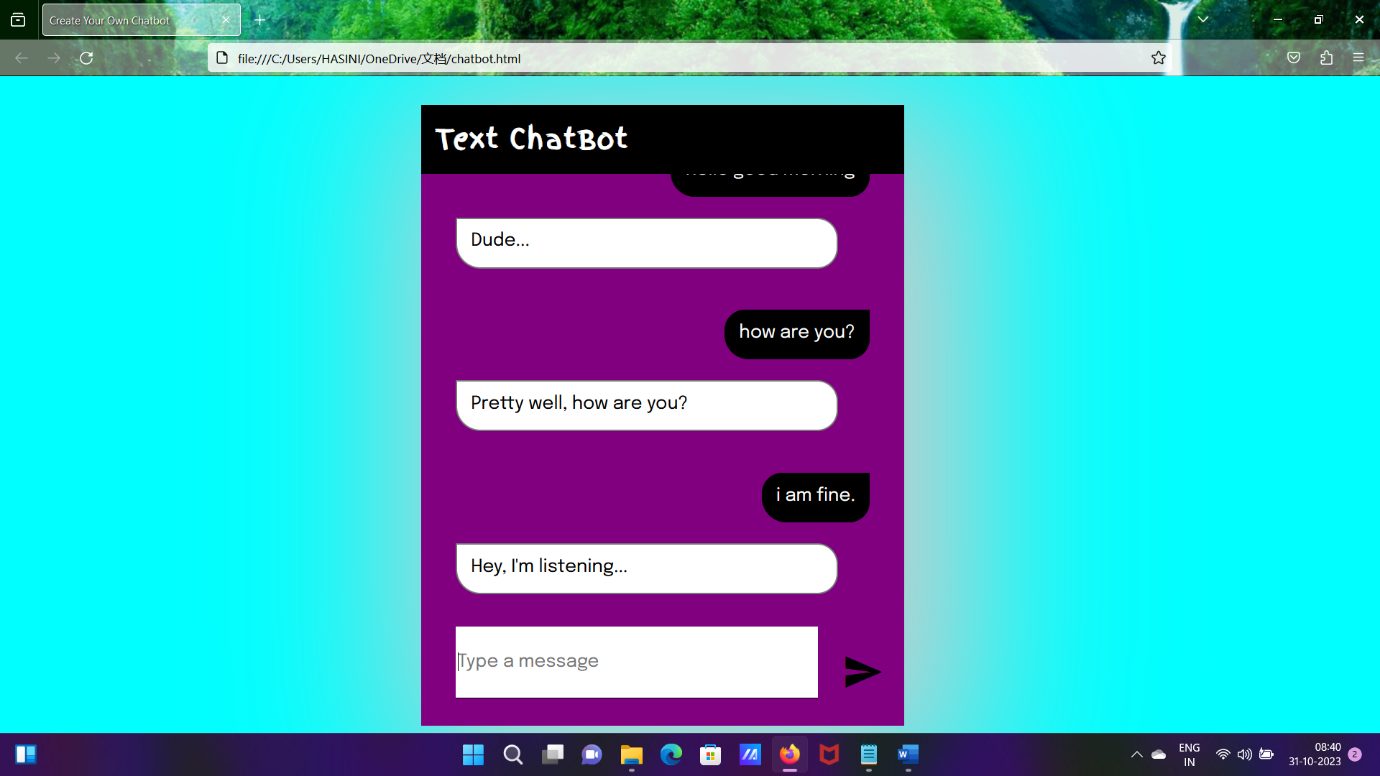
</script>

</body>

</html>

**Output of our text chatbot:**

**** ****

**** ****

**The above mentioned is our chatbot working mechanism.**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Team Members** | **Email ID** |
| **1** | Kayalvizhi K | [kayalvizhik0205@gmail.com](mailto:kayalvizhik0205@gmail.com) |
| **2** | Keerthanasri R | [keerthannasri@gmail.com](mailto:keerthannasri@gmail.com) |
| **3** | Nithyasri R | [nithyasri0329@gmail.com](mailto:nithyasri0329@gmail.com) |
| **4** | Durkavi R | [durkaviramjaya2003@gmail.com](mailto:durkaviramjaya2003@gmail.com) |